ITCS 113 – Fundamentals of Programming
Lecture 1: Extra Exercises

No Need to Submit but Practices (ALL) is encouraged.

Q1: Write a C program to convert from a character into its corresponding ASCII.

- Use `scanf()` to receive a *character* from a user, and use `printf()` to print the output. *Syntax*: `scanf("%c", &input);`
- You may use the ASCII table to check your answer

Sample inputs and output

#	Input	Output	Expected Screen Output
1	`I`	`73`	`I` `73`
2	`c`	`99`	`c` `99`
3	`T`	`84`	`T` `84`
4	`&`	`38`	`&` `38`
5	`9`	`57`	`9` `57`

Q2: Write a C program to convert a number of seconds to hours and print out in the following format: *`hr:min:sec`*

Sample inputs and output

#	Input	Output	Expected Screen Output
1	`33`	`0:0:33`	`33` `0:0:33`
2	`785`	`0:13:5`	`785` `0:13:5`
3	`7402`	`2:3:22`	`7402` `2:3:22`

Q3: Write a C program to calculate the area of a triangle

- Receive a *base* and a *height* of a triangle from a user using `scanf()`
- Calculate the area of the triangle: area = ½ (base*height)
- **Hint**: use `%.2f` for printing *area* with two decimal points.

Sample inputs and output

#	Input	Output	Expected Screen Output
1	3 4	6	3 4 6
2	5.6 33	92.40	5.6 33 92.40
3	8.5 9.7	41.22	8.5 9.7 41.22

Q4: Receive 2 integers, `a` and `b`, from the user, and write a program to generate the `modulo` operation without using %. **Note** that the modulo operation is for finding the remainder after division of `a` by `b`.

Sample inputs and output

#	Input	Output	Expected Screen Output
1	4 2	0	4 2 0
2	5 3	2	5 3 2
3	9 4	1	9 4 1

Q5: The following C code contains bugs. Identify and fix all bugs by

1. Copying these codes into C files and name it `"GiveMeMoney.c"`
2. Compile the files using `gcc` via command line
3. Read the error message, and fix the bugs accordingly.

Hint: There are 12 bugs in the codes.

```
include < studio.h>

int main(
    // Receive an input money
    prntf("Give me your money:");
    int money;
    scanf("%d",money);

    // Change money to coins: 10, 5, 1
    int ten = Money/10;
    int remainMoney = money%10;
    five = remainMoney/5
    remainMoney = remainmoney%5;

    /Display change amounts of each type of coins
    printf("Here is your changes.\n");
    printf("Ten:%d, Five:%d, One:%D\n",ten,&five,remainMoney);
    return 0;
}
```

Sample inputs and output

#	Input	Output	Expected Screen Output
1	23	Here is your change. Ten:2, Five:0, One:3	Give me your money:23 Here is your change. Ten:2, Five:0, One:3
2	77	Here is your change. Ten:7, Five:1, One:2	Give me your money:77 Here is your change. Ten:7, Five:1, One:2

Q6: Write the C program to convert the unit from Kilogram (kg) to Pound (lb.), given 1-kilogram equals to 2.2 pounds. Use `%.2f` for printing output with two decimal points.

Sample inputs and output

#	Input	Output	Expected Screen Output
1	`1`	`2.20`	`1` `2.20`
2	`2.5`	`5.50`	`2.5` `5.50`

Q7: Write the C program to compute $(a-b)^2$ given any input numbers of `a` and `b` respectively. Use `%.2f` for printing area with two decimal points.

Sample inputs and output

#	Input	Output	Expected Screen Output

1	`1`
	`-5.0`
		`36.00`	`1`
			`-5.0`
			`36.00`

2	`-8.0`
	`-8`
		`0.00`	`-8.0`
			`-8`
			`0.00`

Q8: Write the C program to compute the distance of a car traveled in *kilometers* given the speed of a car `s` in *kilometer/hour* (e.g., 80 km/hr) and time a car traveled `t` in *minutes*. Use `%.2f` for printing the output with two decimal points.

Sample inputs and output

#	Input	Output	Expected Screen Output

1	`80`
	`30`
		`40.00`	`80`
			`30`
			`40.00`

2	`120.5`
	`90`
		`180.75`	`-8.0`
			`-8`
			`0.00`

Q9: Write the C program to receive one float numbers (e.g. 42.99) and display the whole number part and the decimal places (the number of digits following the decimal point) separately.

Sample inputs and output

#	Input	Output	Expected Screen Output

1	`42.99`	`Whole number: 42`
		`Decimal places: 99`
			`42.99`
			`Whole number: 42`
			`Decimal places: 99`

2	`87.65`	`Whole number: 87`
		`Decimal places: 65`
			`87.65`
			`Whole number: 87`
			`Decimal places: 65`

Q10: Write the C program to compute the circumference and the area of a circle given an input radius `r` and given PI as a constant of 3.141592. Use `%.2f` for printing the output with two decimal points.

Sample inputs and output

#	Input	Output	Expected Screen Output

1	`1`	`Circumference: 6.28`
		`Area: 3.14`
			`1`
			`Circumference: 6.28`
			`Area: 3.14`

2	`9`	`Circumference: 56.55`
		`Area: 254.47`
			`9`
			`Circumference: 56.55`
			`Area: 254.47`